


[HHSDC Home](#)
[BP Home Page](#)
[The MSC](#)
[CMM](#)
[POST Enterprise](#)
[The Project Office](#)
[Life Cycle Processes](#)
[Search BP](#)
[HHSDC Links](#)
[Resources Library](#)
[QAWG **NEW!**](#)
[SID Policy **NEW!**](#)
[Contact Us](#)


☒ My CA

Functional Testing

Test Main

Test Phases Main

Purpose:

The purpose of functional testing (or component or module testing) is to test small groups of modules that are functionally related. The emphasis is on verifying the interfaces between related modules (inter-module and intra-function), and that utility functions or modules work correctly when called by other modules. This test phase is optional, but recommended for complex processing areas.

Some contracts do not permit State visibility into functional testing (such as when some of the software is COTS). The text below assumes the State does have visibility into this testing phase.

Assumptions/Pre-Conditions:

The contractor/developer should have completed unit testing successfully and all critical errors should have been addressed. An updated version of the code should have been delivered to the Configuration Manager.

Expectations:

- The primary emphasis is the correct passing and setting of parameters (as they pass between the modules), and verification that the design is correctly implemented.
- Functional outputs or module exit values should be verified. This may entail the use of testing tools and debuggers to capture values. The entire range of possible values should be tested to ensure the data is passed and handled correctly between units and modules.
- All error cases should be verified and required to end gracefully with the appropriate error data reported (i.e., handle the error; don't allow the program to terminate on an error). This may include executing re-start logic, recovery from the error, or a graceful shutdown.
- Screen displays and report formats should be verified for format and data accuracy, including appropriate number of decimal places and rounding on monetary values.
- New or modified help screens and supporting user materials should be verified.
- Some performance tests may be conducted and used to model or extrapolate behavior.
- Units should "clean up" after themselves, releasing any system resources, as appropriate. Use test tools to check for "memory leaks" and inefficient processing paths.
- All affected documentation should be updated to reflect fixes and changes, including in-line code comments and unit/module/function headers, design documents, user manuals, help desk procedures or bulletins, and help files. This may be delayed until System Test, depending on change stability and resources.

Responsibilities:

- Creation of Tests - Developer or Tester
- Execution of Tests - Developer or Tester
- Approval of Test Results/Exit Decision (depending on level of State visibility) - Development Manager, Test Manager, QA Manager, Configuration Manager, State Project Manager
- For a complete list of roles and responsibilities, refer to the [Responsibility Assignment Matrix \(RAM\)](#) (MS Word)

Environment:

Development or Test Environment

Type of Data:

Real data (data which was processed on the legacy system and is now being re-used for testing), if possible; else simulated data created to model real data.

Exit Decisions:

- Refer to the [general test exit/acceptance criteria](#).

References:

- IEEE Standard [829-1998](#), Standard for Software Test Documentation (link to pdf)

Samples:

- [CWS/CMS Functional Test Exit Report](#) (MS Word)